

Post-Tests

let

```
Source = Csv.Document(File.Contents("C:\Users\ellie\Downloads\data10.csv"),[Delimiter="," , Columns=30, Encoding=65001,
QuoteStyle=QuoteStyle.None]),
#"Use First Row as Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
#"Change Type" = Table.TransformColumnTypes(#"Use First Row as Headers",{{"No action", Int64.Type}, {"No new comparative
trial", Int64.Type}, {"Prediction Current: same rule", Int64.Type}, {"Prediction Voltage Drop: same rule", Int64.Type}, {"Confidence
Current: verify prediction", Int64.Type}, {"Confidence Voltage Drop: verify prediction", Int64.Type}, {"Rule Current: Confirming
Redundancy", Int64.Type}, {"Rule Voltage Drop: Confirming Redundancy", Int64.Type}, {"Post-test: Current non-normative",
Int64.Type}, {"Post-test: Voltage Drop non-normative", Int64.Type}, {"New comparative trial", Int64.Type}, {"Prediction Current: Fill up
gaps", Int64.Type}, {"Prediction Voltage Drop: Fill up gaps", Int64.Type}, {"Confidence Current: falsify prediction", Int64.Type},
{"Confidence Voltage Drop: falsify prediction", Int64.Type}, {"Identify problem: goal stated", Int64.Type}, {"Rule Current: Simultaneous
scanning", Int64.Type}, {"Rule Voltage Drop: Simultaneous scanning", Int64.Type}, {"Rule Current: Successive scanning",
Int64.Type}, {"Rule Voltage Drop: Successive scanning", Int64.Type}, {"Rule Current: Focus gambling", Int64.Type}, {"Rule Voltage
Drop: Focus gambling", Int64.Type}, {"Rule Current: Conservative Focusing", Int64.Type}, {"Rule Voltage Drop: Conservative
Focusing", Int64.Type}, {"Post-test: Current partial", Int64.Type}, {"Post-test: Voltage Drop partial", Int64.Type}, {"Post-test: Current 1
Valid link", Int64.Type}, {"Post-test: Voltage Drop 1 Valid link", Int64.Type}, {"Post-test: Current 2 Valid links", Int64.Type}, {"Post-test:
Voltage Drop 2 Valid links", Int64.Type}}),
#"Added Custom" = Table.AddColumn(#"Change Type", "Group", each if [Index] <= 20 then "Control"
else if [Index] <= 40 then "Decision Table and Inductive Rules"
else "Inductive Rules"),
#"Changed Type" = Table.TransformColumnTypes(#"Added Custom",{{"Group", type text}}),
#"Added Index" = Table.AddIndexColumn(#"Changed Type", "Index", 1, 1, Int64.Type),
#"Reordered Columns" = Table.ReorderColumns(#"Added Index",{ "Index", "No action", "No new comparative trial", "Prediction
Current: same rule", "Prediction Voltage Drop: same rule", "Confidence Current: verify prediction", "Confidence Voltage Drop: verify
prediction", "Rule Current: Confirming Redundancy", "Rule Voltage Drop: Confirming Redundancy", "Post-test: Current
non-normative", "Post-test: Voltage Drop non-normative", "New comparative trial", "Prediction Current: Fill up gaps", "Prediction
Voltage Drop: Fill up gaps", "Confidence Current: falsify prediction", "Confidence Voltage Drop: falsify prediction", "Identify problem:
goal stated", "Rule Current: Simultaneous scanning", "Rule Voltage Drop: Simultaneous scanning", "Rule Current: Successive
scanning", "Rule Voltage Drop: Successive scanning", "Rule Current: Focus gambling", "Rule Voltage Drop: Focus gambling", "Rule
Current: Conservative Focusing", "Rule Voltage Drop: Conservative Focusing", "Post-test: Current partial", "Post-test: Voltage Drop
```

```

partial", "Post-test: Current 1 Valid link", "Post-test: Voltage Drop 1 Valid link", "Post-test: Current 2 Valid links", "Post-test: Voltage Drop 2 Valid links", "Group")),
  #"Removed Columns" = Table.RemoveColumns(#"Reordered Columns",{ "Group"}),
  #"Added Custom1" = Table.AddColumn(#"Removed Columns", "Group", each if [Index] <= 20 then "Control"
else if [Index] <= 40 then "Decision Table and Inductive Rules"
else "Inductive Rules"),
  #"Changed Type1" = Table.TransformColumnTypes(#"Added Custom1",{{"Group", type text}}),
  #"Removed Other Columns" = Table.SelectColumns(#"Changed Type1",{ "Index", "Post-test: Current non-normative", "Post-test: Voltage Drop non-normative", "Post-test: Current partial", "Post-test: Voltage Drop partial", "Post-test: Current 1 Valid link", "Post-test: Voltage Drop 1 Valid link", "Post-test: Current 2 Valid links", "Post-test: Voltage Drop 2 Valid links", "Group"}),
  #"Changed Type2" = Table.TransformColumnTypes(#"Removed Other Columns",{{"Group", type text}})
in
  #"Changed Type2"

```

Search Strategies

let

```

Source = Csv.Document(File.Contents("C:\Users\ellie\Downloads\data10.csv"),[Delimiter=",", Columns=30, Encoding=65001, QuoteStyle=QuoteStyle.None]),
  #"Use First Row as Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
  #"Change Type" = Table.TransformColumnTypes(#"Use First Row as Headers",{{"No action", Int64.Type}, {"No new comparative trial", Int64.Type}, {"Prediction Current: same rule", Int64.Type}, {"Prediction Voltage Drop: same rule", Int64.Type}, {"Confidence Current: verify prediction", Int64.Type}, {"Confidence Voltage Drop: verify prediction", Int64.Type}, {"Rule Current: Confirming Redundancy", Int64.Type}, {"Rule Voltage Drop: Confirming Redundancy", Int64.Type}, {"Post-test: Current non-normative", Int64.Type}, {"Post-test: Voltage Drop non-normative", Int64.Type}, {"New comparative trial", Int64.Type}, {"Prediction Current: Fill up gaps", Int64.Type}, {"Prediction Voltage Drop: Fill up gaps", Int64.Type}, {"Confidence Current: falsify prediction", Int64.Type}, {"Confidence Voltage Drop: falsify prediction", Int64.Type}, {"Identify problem: goal stated", Int64.Type}, {"Rule Current: Simultaneous scanning", Int64.Type}, {"Rule Voltage Drop: Simultaneous scanning", Int64.Type}, {"Rule Current: Successive scanning",

```

```

Int64.Type}, {"Rule Voltage Drop: Successive scanning", Int64.Type}, {"Rule Current: Focus gambling", Int64.Type}, {"Rule Voltage Drop: Focus gambling", Int64.Type}, {"Rule Current: Conservative Focusing", Int64.Type}, {"Rule Voltage Drop: Conservative Focusing", Int64.Type}, {"Post-test: Current partial", Int64.Type}, {"Post-test: Voltage Drop partial", Int64.Type}, {"Post-test: Current 1 Valid link", Int64.Type}, {"Post-test: Voltage Drop 1 Valid link", Int64.Type}, {"Post-test: Current 2 Valid links", Int64.Type}, {"Post-test: Voltage Drop 2 Valid links", Int64.Type}},
    #"Added Index" = Table.AddIndexColumn(#"Change Type", "Index", 1, 1, Int64.Type),
    #"Reordered Columns" = Table.ReorderColumns(#"Added Index",{"Index", "No action", "No new comparative trial", "Prediction Current: same rule", "Prediction Voltage Drop: same rule", "Confidence Current: verify prediction", "Confidence Voltage Drop: verify prediction", "Rule Current: Confirming Redundancy", "Rule Voltage Drop: Confirming Redundancy", "Post-test: Current non-normative", "Post-test: Voltage Drop non-normative", "New comparative trial", "Prediction Current: Fill up gaps", "Prediction Voltage Drop: Fill up gaps", "Confidence Current: falsify prediction", "Confidence Voltage Drop: falsify prediction", "Identify problem: goal stated", "Rule Current: Simultaneous scanning", "Rule Voltage Drop: Simultaneous scanning", "Rule Current: Successive scanning", "Rule Voltage Drop: Successive scanning", "Rule Current: Focus gambling", "Rule Voltage Drop: Focus gambling", "Rule Current: Conservative Focusing", "Rule Voltage Drop: Conservative Focusing", "Post-test: Current partial", "Post-test: Voltage Drop partial", "Post-test: Current 1 Valid link", "Post-test: Voltage Drop 1 Valid link", "Post-test: Current 2 Valid links", "Post-test: Voltage Drop 2 Valid links"}),
    #"Added Custom" = Table.AddColumn(#"Reordered Columns", "Group", each if [Index] <= 20 then "Control"
else if [Index] <= 40 then "Decision Table and Inductive Rules"
else "Inductive Rules"),
    #"Changed Type" = Table.TransformColumnTypes(#"Added Custom",{{"Group", type text}}),
    #"Removed Columns" = Table.RemoveColumns(#"Changed Type",{"Post-test: Current 1 Valid link", "Post-test: Current 2 Valid links", "Post-test: Current non-normative", "Post-test: Current partial", "Post-test: Voltage Drop 1 Valid link", "Post-test: Voltage Drop 2 Valid links", "Post-test: Voltage Drop non-normative", "Post-test: Voltage Drop partial"})
in
    #"Removed Columns"

```

Engineering Approaches

```

let
    Source = Csv.Document(File.Contents("C:\Users\ellie\Downloads\data8.csv"),[Delimiter="#(tab)", Columns=19, Encoding=1252, QuoteStyle=QuoteStyle.None]),

```

```

#"Use First Row as Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
#"Change Type" = Table.TransformColumnTypes("#Use First Row as Headers",{{"No action", Int64.Type}, {"1 bulbs equal",
Int64.Type}, {"2 bulbs equal", Int64.Type}, {"1 vs. 2 bulbs different", Int64.Type}, {"Battery voltage equal", Int64.Type}, {"Battery
voltage different", Int64.Type}, {"Resistance change: 1 or 2 bulbs", Int64.Type}, {"Resistance change: 2 bulbs", Int64.Type},
{"Resistance equal: 2 bulbs different value", Int64.Type}, {"Resistance equal: 2 or 1 vs. 2 bulbs same value", Int64.Type},
{"Resistance equal: 1 bulb", Int64.Type}, {"Resistance equal: 2 bulbs different value & position", Int64.Type}, {"Ammeter equal:
after/before 1 or 2 bulb", Int64.Type}, {"Ammeter different: before vs. after 1 or 2 bulbs", Int64.Type}, {"Ammeter different: on high/low
of 2 bulb", Int64.Type}, {"Voltmeter equal: 1 bulb or 2 bulbs or battery", Int64.Type}, {"Voltmeter equal: 2 bulbs", Int64.Type},
{"Voltmeter different: high vs. low resistance 2 bulbs", Int64.Type}, {"Voltmeter different: high vs. low resistance 2 bulbs vs. battery",
Int64.Type}}),
#"Added Custom" = Table.AddColumn("#Change Type", "Group", each if [Index] <= 20 then "Control"
else if [Index] <= 40 then "Decision Table and Inductive Rules"
else "Inductive Rules"),
#"Changed Type" = Table.TransformColumnTypes("#Added Custom",{{"Group", type text}}),
#"Added Index" = Table.AddIndexColumn("#Changed Type", "Index", 1, 1, Int64.Type),
#"Reordered Columns" = Table.ReorderColumns("#Added Index",{{"No action", "1 bulbs equal", "2 bulbs equal", "1 vs. 2 bulbs
different", "Battery voltage equal", "Battery voltage different", "Resistance change: 1 or 2 bulbs", "Resistance change: 2 bulbs",
"Resistance equal: 2 bulbs different value", "Resistance equal: 2 or 1 vs. 2 bulbs same value", "Resistance equal: 1 bulb",
"Resistance equal: 2 bulbs different value & position", "Ammeter equal: after/before 1 or 2 bulb", "Ammeter different: before vs. after
1 or 2 bulbs", "Ammeter different: on high/low of 2 bulb", "Voltmeter equal: 1 bulb or 2 bulbs or battery", "Voltmeter equal: 2 bulbs",
"Voltmeter different: high vs. low resistance 2 bulbs", "Voltmeter different: high vs. low resistance 2 bulbs vs. battery", "Index",
"Group"}),
#"Removed Columns" = Table.RemoveColumns("#Reordered Columns",{{"Group"}}),
#"Reordered Columns1" = Table.ReorderColumns("#Removed Columns",{{"Index", "No action", "1 bulbs equal", "2 bulbs equal", "1
vs. 2 bulbs different", "Battery voltage equal", "Battery voltage different", "Resistance change: 1 or 2 bulbs", "Resistance change: 2
bulbs", "Resistance equal: 2 bulbs different value", "Resistance equal: 2 or 1 vs. 2 bulbs same value", "Resistance equal: 1 bulb",
"Resistance equal: 2 bulbs different value & position", "Ammeter equal: after/before 1 or 2 bulb", "Ammeter different: before vs. after
1 or 2 bulbs", "Ammeter different: on high/low of 2 bulb", "Voltmeter equal: 1 bulb or 2 bulbs or battery", "Voltmeter equal: 2 bulbs",
"Voltmeter different: high vs. low resistance 2 bulbs", "Voltmeter different: high vs. low resistance 2 bulbs vs. battery"}),
#"Added Custom1" = Table.AddColumn("#Reordered Columns1", "Group", each if [Index] <= 20 then "Control"
else if [Index] <= 40 then "Decision Table and Inductive Rules"
else "Inductive Rules"),

```

```
#"Changed Type1" = Table.TransformColumnTypes("#Added Custom1",{{"Group", type text}})
in
#"Changed Type1"
```